Light Pen Manual

for use with apple][computers

SYMTEC INC.

ABW CORP.
software developed by

SYMTEC LIGHT PEN REFERENCE MANUAL

Copyright (c) 1978 by
ABW Corp. and Symtec Inc.

TABLE OF CONTENTS

I. INITIAL SET UP 1

Installing the Symtec Light Pen. 1
Alligning Your Pen 2
Pen Head Sensitivity Adjustment. 7
Running the Demos. 8

II. USING THE PROVIDED SYMTEC LIGHT PEN SUBROUTINES. . . 10

Table of Cassette Contents 10
What the Subroutines Can Do. 11
Linking the Subroutines to Your Program. 22

III. UNDERSTANDING THE PEN SUBROUTINES. 23

Understanding the Screen Modes 23
Theory of Pen Operation. 25
Text Outputting and the Non-Destructive Cursor . . . 30

IV. APPENDICES 34

I. INITIAL SET UP

Installing the Symtec Light Pen

The Symtec Light Pen comes in two basic parts. The PC card and the Light Pen Head Assembly. These are connected by a coiled telephone-type cord. To install the PC card, take the following steps:

1. Turn the Apple II off.
2. Open lid on Apple II case.
3. Install PC card in slot #5 (see Figure 1).

   NOTE: Slot #5 is the sixth slot from the left when seated in front of the keyboard. When installing PC card, components are to the right as you sit in front of the keyboard.
4. Clip alligator lead to center post of video output jack labelled K-14 in Figure 1.
5. Turn video output potentiometer on Apple II board fully clockwise.
6. Dress Head Assembly cord through rear of Apple II case.
7. Leave lid off and turn power on.

You are now ready to align the sensitivity of your pen and register it to your individual television set.

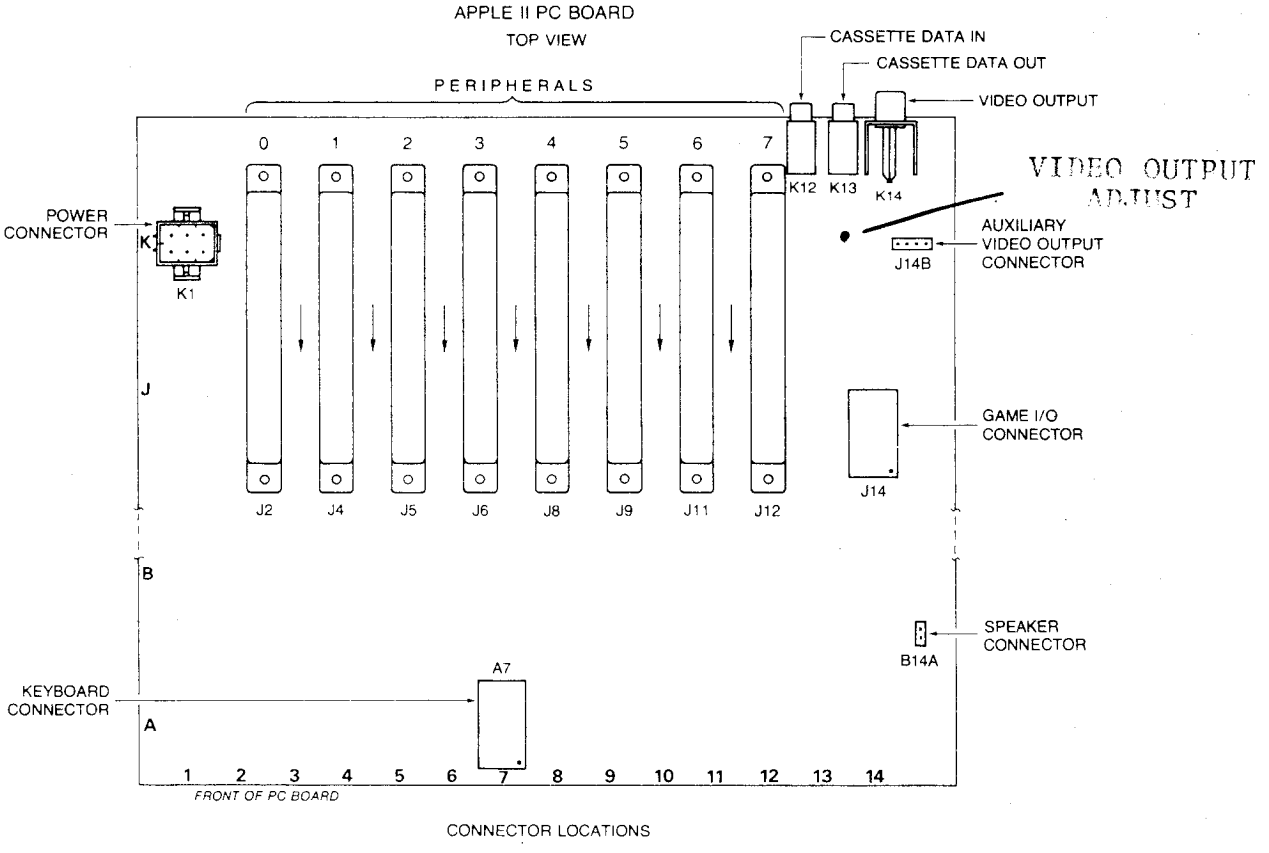


Figure 1

Alligning Your Pen

To allign your Symtec Light Pen you will need to load the registration program on cassette side #1. Do this as follows:

1. Rewind tape.
2. Hit reset button on keyboard. An asterisk (*) will appear.
3. Type C00.FFFR

4. Play tape.
5. Hit RETURN.
6. When asterisk reappears, tape has finished. Hit STOP key of tape recorder.
7. Depress and hold the CONTROL key. Hit the B key and release the CONTROL key. Hit the RETURN key; this will give you a prompting character of >. This puts you in the correct mode to load a BASIC program.
8. Type LOAD.
9. Play tape from present position.
10. Hit RETURN key. When prompt character returns and computer beeps, program has been loaded.

NOTE: This is the exact routine used to load the HIRES demo tape listed on page 15 of your Apple II Reference Manual.

Now that you have your registration program loaded, run the program in the usual method. A point will appear in each corner of the screen. Point the pen at any one of the dots and position it until the computer beeps continuously. If no beeps occur, increase the screen brightness. (If your television set is adjusted to maximum brightness and you are unable to obtain any beeping whatsoever, use the procedures listed under Sensitivity Adjustment in this section.) The computer should beep about once per second. If it doesn't,

refer again to the sensitivity adjustment procedures. Do not adjust the pen head sensitivity potentiometer, located on the body of the pen head, if the computer is beeping properly. It has been pre-set at the factory and unless you are encountering difficulty, should not be adjusted.

Point the pen at the middle of the screen where there are no dots; the beeping should stop. If the beeping continues, lower the screen brightness until it does. Now point to one of the four dots again, the beeping should resume. Look at the numbers labeled "X," "Y," and "ZQ" at the bottom of the screen. The value indicated under ZQ is a program check. If it is consistently greater then 5, the light pen is not functioning properly. The other two numbers represent the X and Y coordinates of the dot you are touching, in the HIRES coordinate system. The light pen must be registered to your television set so that these coordinates, as well as those of the remaining three dots, agree with those shown in Figure 2.

To register the pen, locate the three potentiometers on the top of the PC board (see Photo 1). You will notice one of the potentiometers is a different color and has a drop of sealing adhesive on it. Do not adjust this potentiometer unless you are attempting to sync off a signal which is other than that provided by the Apple II. The two other potentiometers will be labeled X and Y respectively. They control the X and Y registration of the light pen with respect to your

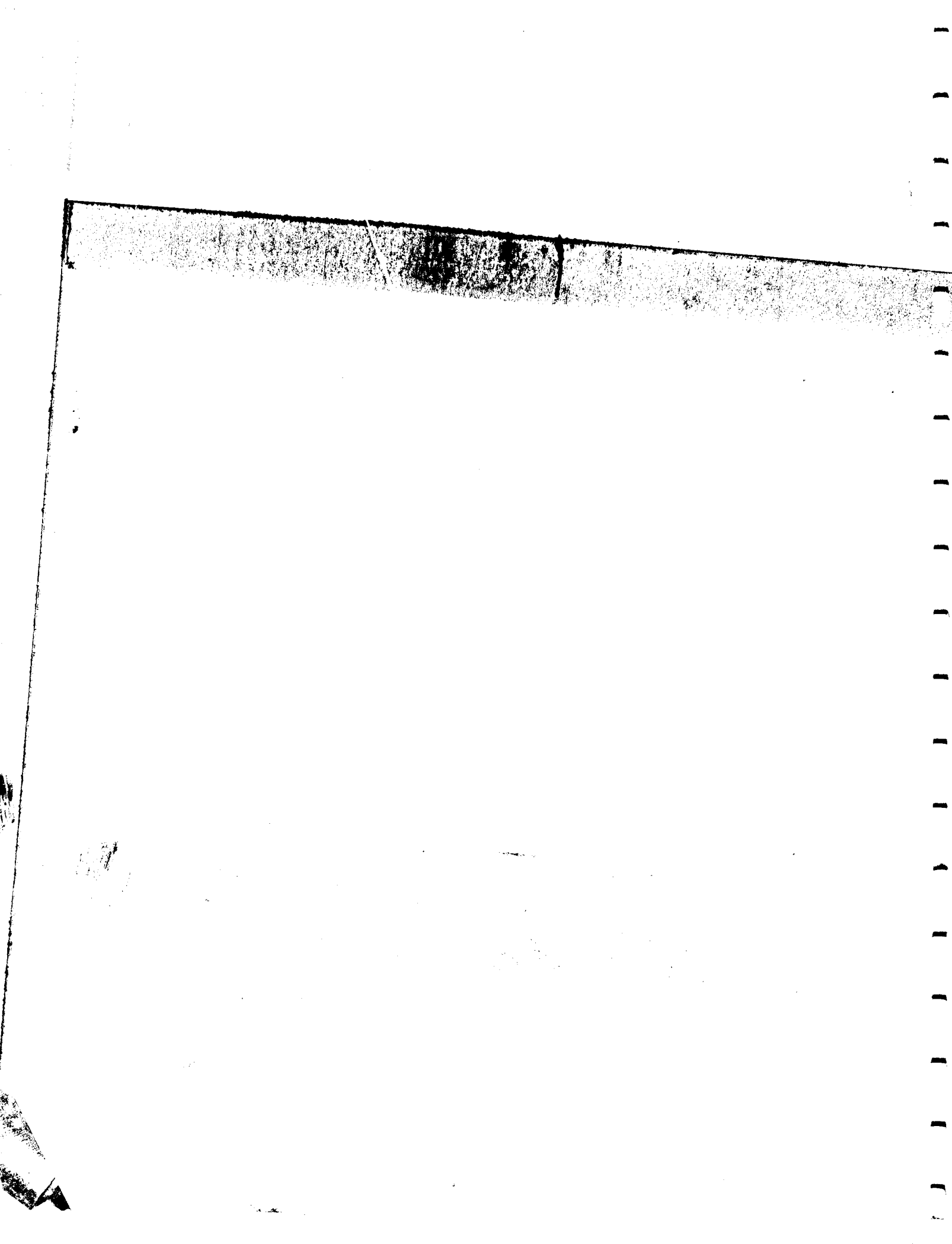
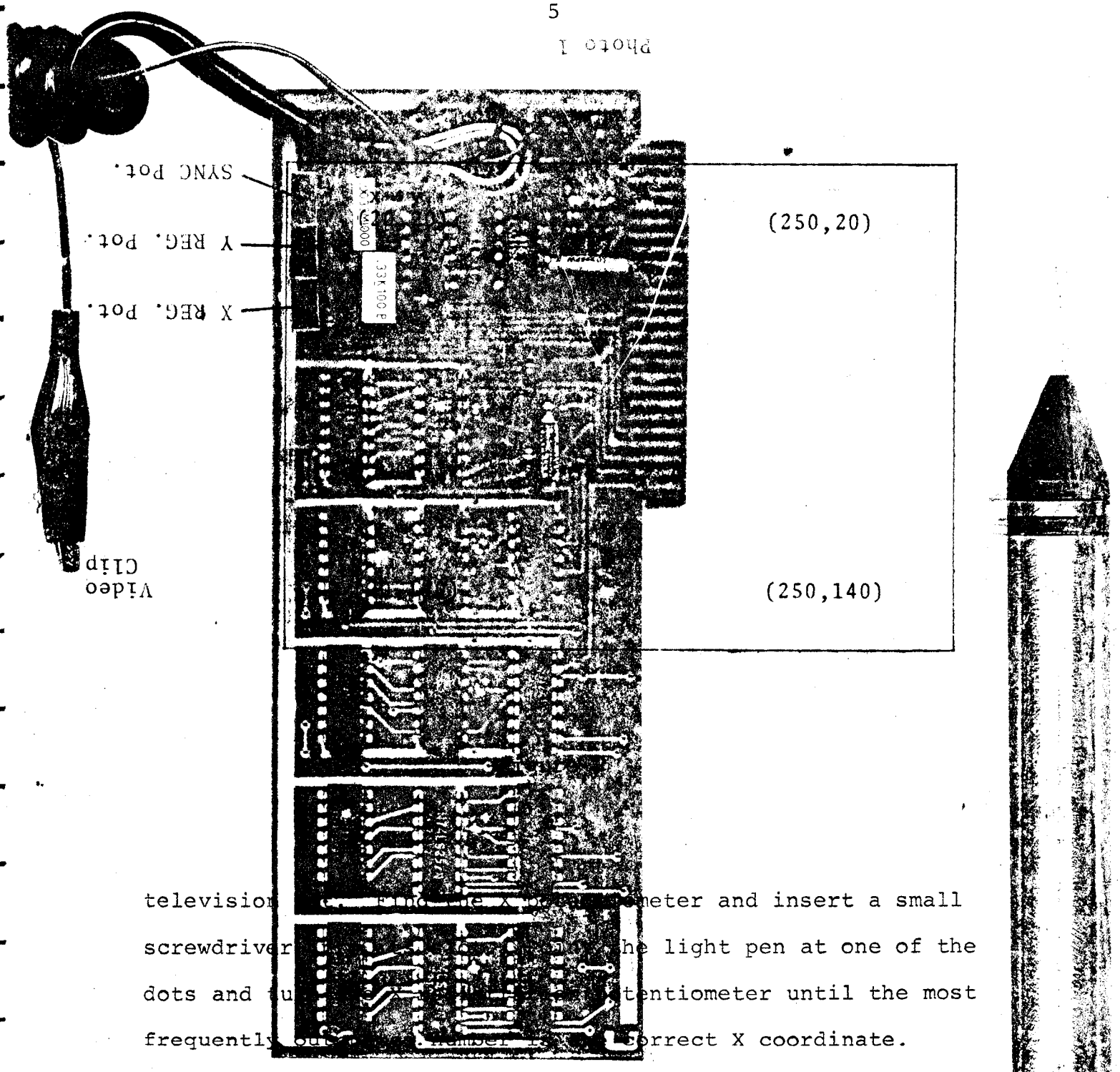


Photo 1



television [illegible] meter and insert a small screwdriver [illegible] the light pen at one of the dots and [illegible] tentiometer until the most frequently [illegible] correct X coordinate. (There will be a normal oscillation of ± one from this most frequent value.) Move the light pen to the other three points and make sure that they also have the correct X coordinate. Repeat this procedure with the Y coordinate. When this has been completed, you should be able to point the light pen at

any one of the four points and obtain a coordinate which is within one of the correct HIRES coordinate of that point. You are now in registration with your television set. Touch and hold ring while pointing pen head at screen to exit program. Should you use your light pen with another television receiver, it will have to be re-registered. You are now ready to test the pen, turn to the section labeled Demos to run the demo programs supplied with your Symtec Light Pen.

NOTE: It was required for this procedure that the light pen not return a HIRES value when pointed at a black part of the screen. In your own program you may want the light pen to return a value when it is pointed at a dark portion of the screen. To do this, simply increase the screen brightness on your television receiver until this is accomplished. There are few receivers which cannot be made bright enough to read a "dark" point.

Pen Head Sensitivity Adjustment

Locate the pen sensitivity potentiometer on the body of the Pen Head Assembly (see Photo 1). You will need to adjust this using a small jeweler's type screwdriver. Increase the screen brightness to maximum and point the pen at one of the dots in the four corners of the screen provided by the registration program. Turn the sensitivity potentiometer counterclockwise to its maximum setting. Slowly increase the potentiometer in the clockwise direction until a beeping is

heard. The X coordinate value should be significantly less than the actual value of X. If this is not the case, return the sensitivity potentiometer to its maximum counterclockwise position and repeat this procedure. Once you have obtained this condition, very slowly advance the sensitivity potentiometer in the clockwise direction. You should then obtain X readings which are very close to the actual value of the X coordinate. Point the pen at the middle of the screen where there are no dots. Decrease your screen brightness until the beeping is no longer heard. Now point the pen at one of the four dots and the beeping should resume. Continue rotating the sensitivity potentiometer in the clockwise direction until the beeping stops once again. This defines a zone of potentiometer settings which output a reasonable value of X. The correct and optimal setting for its sensitivity is approximately in the middle of that zone so that the X output is consistently at a minimum. Once your pen sensitivity has been optimized, restart the registration procedures previously listed.

Running the Demos

Following the registration program on cassette side #1 is the Light Pen Demo routine (the third file on that side). Load this as you would any other INTEGER BASIC program. This program will demonstrate some of the capabilities of the Symtec Light Pen. There are two other demo programs provided: A one and two man checkers routine on cassette #5, and a

catalog menu select feature for use with the Apple II Disc and Disc Operating System. The latter program allows you to use the light pen to select and run a program listed in the catalog. To do so, load this routine onto your disc and save it using the name "C." Now, by issuing the command RUN C, the screen will clear, the catalog will be listed, and the light pen will have a non-destructive cursor following where it points. To select a program, position the non-destructive cursor on the first letter of the file name you wish to run and touch the ring on Light Pen Head Assembly. That program will execute immediately. The menu select feature cannot be used with catalogs that exceed one screen in length. Both the checkers and the catalog select demos are INTEGER BASIC and must be loaded as such, with the checkers routine requiring 16K of memory.

II. USING THE PROVIDED SYMTEC LIGHT PEN SUBROUTINES

Table of Cassette Contents

Cassette Side #	Name of Program	Language	Special Requirements
1	Registration Program	Machine code and INTEGER BASIC	No special requirements
	Light Pen Demo	Machine language and INTEGER BASIC	No special requirements
2	HIRES Numbers	Machine code	No special requirements
3	HIRES Numbers	INTEGER BASIC	No special requirements
	User Mode Select Numbers	INTEGER BASIC	No special requirements
4	HIRES Numbers	Applesoft II*	No special requirements
	User Mode Select Numbers	Applesoft II	No special requirements
5	One or Two Man Checkers	INTEGER BASIC	16K
6	Catalog Selector	INTEGER BASIC	DOS

*Tape Version

What the Subroutines Can Do

The HIRES number subroutines, whether in machine code, INTEGER BASIC, or Applesoft II basic, do exactly the same thing--return a HIRES number corresponding to where the pen head is pointing on the screen independent of the screen mode, i.e., LORES, HIRES, or text, that you are in at the present time. Let's first talk about the INTEGER BASIC or Applesoft II basic versions of the subroutines. Both versions of the HIRES number subroutine use the same procedure to find the X and Y coordinates of the light pen position. They start at line 29000 and end at line 29099. These high line numbers were given to the subroutines so that they may be inserted into existing programs without conflict. A GOSUB to these subroutines will return to you the HIRES X coordinate in the variable ZX and the HIRES Y coordinate in the variable ZY. All variables used in each subroutine are prefixed by the letter Z. Again, to distinguish them from variables external to the subroutine. The other variables used in these subroutines are Z1, Z2, Z3, Z4, ZA, and ZQ. Be sure not to use these variables in other places outside the subroutines.

Example 1 on the next page illustrates how the HIRES subroutine could be used. From the main program we simply GOSUB 29000 and take the X and Y coordinates returned from the subroutine in ZX and ZY and print them. If we wanted the coordinates in another system other than HIRES, we can convert them by using the algorithm supplied in Appendix 2.

```
>LIST
   10 TEXT : CALL -936: GR : COLOR=
      15
   20 FOR I=0 TO 39: HLIN 0,39 AT
      I: NEXT I
   30 GOSUB 29000
   40 VTAB 21: PRINT "  X=";ZX;"     Y=
      ";ZY;"     "
   50 GOTO 30
29000 REM  LIGHT PEN SUBROUTINE
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=0 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256*( PEEK
      (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN RETURN

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN
29100 REM     ##################
29101 REM     #                #
29102 REM     # COPYRIGHT (C)  #
29103 REM     #                #
29104 REM     #      1978      #
29105 REM     #                #
29106 REM     #       BY       #
29107 REM     #                #
29108 REM     #    ARW CORP    #
29109 REM     #                #
29110 REM     #      AND       #
29111 REM     #                #
29112 REM     #   SYMTEC INC   #
29113 REM     #                #
29114 REM     ##################
```

Example 1

The machine code version of this routine also returns the X and Y coordinates in HIRES. This program is loaded using the command 4000.407FR and it is loaded in the same manner as the first part (C00.FFFR) of the registration program. This puts the machine code version in memory location starting at the Hex code number 4000 and ending at the Hex code address 407F. If these locations in memory are occupied, you may put it at another location in memory, provided it has length of 70 in Hex. This subroutine is accessed by making a CALL 16384. After that has been completed, in memory location 4078 in Hex or 16504 in decimal will be the HYRES Y coordinate, in 16501 or 4075 in Hex will be the HIRES X coordinate. In 16500 or 4074 in hex will be a number that, if it is greater than 127, signifies that you should add 256 to the X coordinate. This number shows whether or not the X coordinate is greater than 255, the largest number that can be represented by one byte of memory. These numbers can be accessed by a PEEK with the appropriate decimal address. If you are moving the machine program to another location in memory, make sure that when you PEEK to get your X and Y coordinates, you PEEK at an address which is different from the addresses given by the same amount that the start of where you are loading the machine program is different from 4000 Hex. To illustrate how the subroutine may be used, Example 2 shows how to do exactly what was done with Example 1 except with the machine code program.

We have discussed how to use the subroutines which

```
>LIST
   10 TEXT : CALL -936: GR : COLOR=
      15
   20 FOR I=0 TO 39: HLIN 0,39 AT
      I: NEXT I
   30 CALL 16384
   35 ZX= PEEK (16501)+256*( PEEK
      (16500)>127)
   37 ZY= PEEK (16504)
   40 VTAB 21: PRINT "  X=";ZX;"     Y=
      ";ZY;"     "
   50 GOTO 30
```

Example 2

return a HIRES number corresponding to the pen location. If you are not in HIRES mode, it is necessary to convert them using the algorithms in Appendix 2. If the operator does not desire to convert them, we have subroutines provided which do the conversion automatically. Plus, they allow the user, when in text mode, to see a non-destructive cursor which follows the pen. The non-destructive cursor indicates where the pen is pointed by flashing the letter or blank space of text which the pen is pointed at. As the pen moves across text, this non-destructive cursor will travel with it, not erasing any of the text as it goes, hence its name. These subroutines can be made so that, in the text mode, when the pen touch ring is activated, the subroutine will return not only the coordinate of the text character that you are looking at, but also an alpha string of length one which has in it the character indicated by the pen. Thus, the light pen and these sub-routines can be used exactly to correspond with a normal one character INPUT from the keyboard. This is very useful in substituting these subroutines in existing programs that were designed for keyboard entry. Let us now explain how to use these new subroutines.

The Applesoft II basic and INTEGER BASIC versions of the subroutines are approximately the same with one significant difference. This difference will be elaborated upon later in the discussion. Again, the X and Y coordinates of pen position are assigned to variables ZX and ZY. And all variables in the

program are prefixed by the letter Z. These variables are ZA, ZI, ZM, ZN, ZP, ZQ, ZT, ZU, ZV, ZW, ZX, ZY, Z1, Z2, Z3, Z4, Z9, plus the alpha variables ZR$, ZZ$. Most of these variables will not be accessed by any part of the program external to the subroutine. And unless otherwise indicated, should not be defined anywhere else in your program. The variables that will be accessed external to the subroutines are of course, ZX and ZY plus the ZM, ZP, ZT, ZZR and the INTEGER BASIC version ZR$. These variables are called arguments of the subroutine and must be passed an appropriate value. These arguments are defined as follow:

Name	Description
ZX	This gives the X coordinate of pen position in the screen mode which you have selected.
ZY	This gives the Y coordinate of pen position as above.
ZM	This tells the subroutine what mode the screen is in. The following modes are possible: ZM = 0 Screen is in full screen HIRES mode. ZM = 1 Screen is in HIRES mode with four lines of text at bottom. ZM = 2 Screen is in full screen LORES mode. ZM = 3 Screen is in LORES mode with four lines of text at bottom. ZM = 4 Screen is in full screen text mode.
ZT	If ZT = 1, the alpha string ZZ$, length one, will be set equal to the text character indicated by the pen. If ZT = 0, this will not happen.

Name	Description
ZP	This argument tells you whether or not in a mixed graphic mode, i.e., HIRES with four lines of text on bottom or LORES with four lines of text on bottom, if you are in the text portion of the screen. If ZP = 1, then you are in the four lines of text at the bottom of the screen. If ZP = 0, you are in the other portion of the screen.
ZZ$	This is an alpha string, length one which if ZT = 1 will equal the text character at which the pen is pointed.
ZR$	This argument is only used in the INTEGER BASIC version of the subroutine. It is an alphabet string which is passed to the subroutine so that it can detect which character the pen is positioned over.

An important point to be made at this time is that the user mode select subroutines all use the pen touch ring to select a portion of the screen. When you have positioned the pen to the correct setting on the television screen, you simply touch the ring until a beep is heard. The subroutine will then respond with the correct X and Y coordinates in the screen mode that you have indicated by the argument ZM. The HIRES number routines previously discussed did not use the touch ring on the Light Pen Head Assembly. As soon as they detected a point on the screen, they returned that coordinate in HIRES from the subroutine. This made the touch ring independent of the previous HIRES number subroutine, i.e., whereas it is active in the user mode select number subroutine. To access the

touch ring, one simply PEEKS at location -16170. If this number is greater than 127, then the touch ring is being touched. If it is not greater, then it is not being touched. Good electrical contact must be made with the thumb or finger of the operator.

Let us illustrate how to use the user mode select subroutine by a few examples. Example 3 shows a main program which, by setting ZM = 3, indicates that we are in the mixed mode between LORES graphic and four lines of text at the bottom of the screen. When we GOSUB 29000 with ZM = 3 and ZT = 0 we will get a LORES number with X and Y coordinates between 0 and 39 if the pen is pointing to the LORES part of the screen when the button is depressed and the beep is heard. This will be indicated by ZP being equal to 0. If the pen is pointing at the text portion of the screen when the touch ring is depressed, then ZP will equal 1 when the subroutine returns. This tells us that we are in the text portion of the screen. The X coordinate in text mode will go from 1 to 40. This corresponds to the range of the TAB function and the Y coordinate will be from 21 to 24. Again, corresponding to the VTAB function. If you TAB ZX and VTAB ZY, your next printed output will begin at the exact position that the pen had selected. If we had set ZT = 1, then the character which we were pointed at in the text portion of the screen would be returned by the alpha string ZZ$. In this way an input to the keyboard of one letter, let us say Y or N for yes or no in a

```
LIST 1,29099
    1 DIM ZZ$(1),ZR$(64)
    2 ZR$="@ABCDEFGHIJKLMNOPQRSTUVWXYZ
      ZZZ↑Z !Z$$%&'( )*+,-./0123456789:
      ;<=>?"
    3 ZT=0:ZM=3
    4 GR : COLOR=15
    5 FOR I=0 TO 39
    6 FOR J=0 TO 39
    7 PLOT I,J
    8 NEXT J
    9 NEXT I
   14 GOSUB 29000
   15 PRINT ZX,ZY,ZP
   16 GOTO 14
29000 REM  LIGHT PEN SUBROUTINE
29003 ZV=0
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=1 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256*( PEEK
      (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN 29100

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN

29100 REM  NONDESTRUCT CURSOR AND TEXT

29103 IF ZM$4 AND (ZM$3 OR ZY<160
      ) AND (ZM$1 OR ZY<160) THEN
      29129
29106 IF ZV$0 THEN POKE ZW,Z9
29109 ZU=1024+128*((ZY/8) MOD 8)+
      40*(ZY/64)+ZX/7
29112 ZN= PEEK (ZU)
29115 IF ZT=0 THEN 29124
29118 ZS=(ZN MOD 64)+64*((ZN MOD
      64)<32)+128
29121 FOR ZI=1 TO 64: IF ASC(ZR$(
      ZI,ZI))$ZS THEN NEXT ZI
29122 ZZ$=ZR$(ZI,ZI)
29124 POKE ZU,((ZN MOD 64)+64)
29125 FOR ZI=1 TO 100: NEXT ZI
29127 ZW=ZU:ZV=1:Z9=ZN
29128 GOTO 29130
29129 IF ZV=0 THEN 29130: POKE ZW,
      Z9:ZV=0
29130 IF PEEK (-16170)<128 THEN 29005

29133 IF ZM$4 AND (ZM$3 OR ZY<160
      ) AND (ZM$1 OR ZY<160) THEN
      29136
29134 POKE ZW,Z9
29135 REM  MODE COMPENSATION
29136 ZP=0
29139 IF ZM$1 THEN 29154
29142 IF ZY<160 THEN 29154
29145 ZY=ZY/8+1
29148 ZX=ZX/7+1
29151 ZP=1
29154 IF ZM$2 AND ZM$3 THEN 29175

29157 ZX=ZX/7
29160 ZY=ZY/4
29162 CALL -1052
29163 IF ZM=2 OR ZY<40 THEN 29175

29166 ZY=ZY/2+1
29169 ZX=ZX+1
29172 ZP=1
29175 IF ZM$4 THEN 29184
29178 ZX=ZX/7+1
29181 ZY=ZY/8+1
29184 CALL -1052
29187 RETURN
```

Example 3

portion of a program, could be redone with the light pen by simply saying ZP = 1, ZM = 3 if we were in the mixed LORES text mode--GOSUB 29000 and let the character we were inputting, let us say ANF$ for answer$ = ZZ$. This will correspond exactly to the statement input ANF$, i.e., when the input was looking for one character, either Y or N to corresponding to yes or no.

The importance of ZP must be stressed. Going back to our example, if we are in mixed mode between and LORES and text, it is possible to have return the value 1, 21 for both a position on the LORES part of the screen and a position on the text part of the screen. There will be an ambiguity if not for the ZP argument. Let us now discuss the difference between the Applesoft II and INTEGER BASIC version of the subroutine. You will note that it is necessary to dimension the ZZ$ variable to length one outside the routine. This is done so that on subsequent GOSUBS to the subroutine we will not redimension the variable and incur an error. In INTEGER BASIC there is no function which goes from the ASCII representation of a character to the character itself, i.e., such as, CHR$ does in Applesoft II. For this reason, we must pass the subroutine an alphabet string so that it may use the alphabet string to identify the character which the pen is looking at. In the INTEGER BASIC version, therefore, it is necessary to dimension another alpha string, ZR$ to length 64 and initialize it as in line 2 of the example. Note that this

initialization does not allow any characters which are not enterable from the keyboard plus it does not allow the character ". If those characters are encountered, the alpha string ZZ$ will be equal to the letter Z. The methods which are used to identify which character the pen is looking at and generate the non-destructive cursor are discussed in Section III.

Let us provide a quick recap for this section. The HIRES number subroutine both in INTEGER BASIC and Applesoft II return to you only the HIRES coordinate of the pen location. From the listing at the end of this manual, it can be seen that the HIRES number subroutines are much smaller than the user mode select subroutines and therefore they execute much quicker. But once the subroutine has been accessed by a GOSUB it will return the coordinate of the first position that the pen sees. It is independent of the TOUCH ring on the Light Pen Assembly. The user mode select subroutines are longer, they are somewhat slower to run and use and they will wait until you select a position by using the TOUCH ring on the Light Pen Head Assembly and hearing a beep. In the INTEGER BASIC version of the user mode select subroutine, you must dimension ZR$ to 64 and initialize it with an alphabet string as in Example 3. This is not necessary in the Applesoft II version, but it is necessary to dimension ZZ$ to length 1. The argument, ZM, ZP, ZT, ZZ$ (ZR$ in the INTEGER BASIC case), ZX and ZY are used external to the subroutine. The arguments

ZM, ZT, must be initialized before going to the subroutine. When in the text mode, or in the text portion of mixed mode, either LORES text of HIRES text, a non-destructive cursor will follow the pen.

Linking the Subroutines to Your Program

To link the subroutines provided, whether the HIRES number or user mode select number subroutine, it is necessary to load them from the cassette tape on which they are located and build your program around them. The programs on cassette tapes cannot be loaded into an existing program. If you desire to use them with an existing program, they must be entered via the keyboard. Both subroutines use line numbers that begin with 29000 and variables that begin with the letter Z, so as not to interfere with existing programs. If you desire to use the machine code version, that can be loaded in directly without disturbing existing programs, provided that the location selected for the machine code subroutine is clear in memory.

III. UNDERSTANDING THE PEN SUBROUTINES

Understanding the Screen Modes

Basically there are three different screen modes and two modes which are mixed version of these three modes. The three modes are: 1) HIRES mode, this has 280 locations (pixels) in the X direction, labelled 0 to 279 and 192 pixels labelled 0 to 191 in the Y direction. This is for full screen HIRES. 2) LORES mode, this has 40 pixels in the X direction labelled 0 to 39 and 48 pixels in the Y direction labelled 0 to 47. Again, this is the full screen LORES mode. 3) Text, this has 40 pixels in the X direction labelled 1 to 40 and 24 pixels in the Y direction, labelled 1 to 24. In the two mixed modes there are: 1) HIRES/Text mode. This has the HIRES portion of the screen with 280 X pixels numbered 0 through 279 and 160 pixels in the Y direction labelled 0 through 159. It also has four lines of text at the bottom of the screen so that there are four pixels of text in the Y direction labelled 21 through 24 and 40 pixels in the X direction labelled 1 through 40. 2) LORES/Text mode. In the LORES portion of the screen, which is the upper portion there are 40 pixels in the X direction labelled 0 through 39 and 40 pixels in the Y direction labelled 0 through 39. There again are four lines of text on the bottom of the screen which mean there are 4

pixels in the Y direction labelled 21 through 24 and 40 pixels in the X direction labelled 1 through 40.

For a pictorial representation of these different screen modes see Appendix 2. To convert from one mode to another we must define the LORES pixel and the text pixel in terms of the HIRES pixel. The LORES pixel which is the square that you would see on your television screen if you plotted a single LORES point is actually 4 pixels of HIRES in the Y or vertical direction and 7 pixels of HIRES in the X or horizontal direction. The text pixel is 8 HIRES pixels in the Y or vertical direction and 7 HIRES pixels in H or horizontal direction. Notice that one text pixel is 2 LORES pixels. So now to convert from HIRES to any other screen mode is simply an algebraic algorithm. These are listed in equation 1 through 4. Notice that the text portion of a mixed mode screen is exactly the same coordinates as the text portion on a full text screen and also the LORES portion of a mixed mode screen is exactly the same as that of a full screen LORES representation.

Equation

1 X TEXT = (X HIRES/7) + 1

2 Y TEXT = (Y HIRES/8) + 1

3 X LORES = X HIRES/7

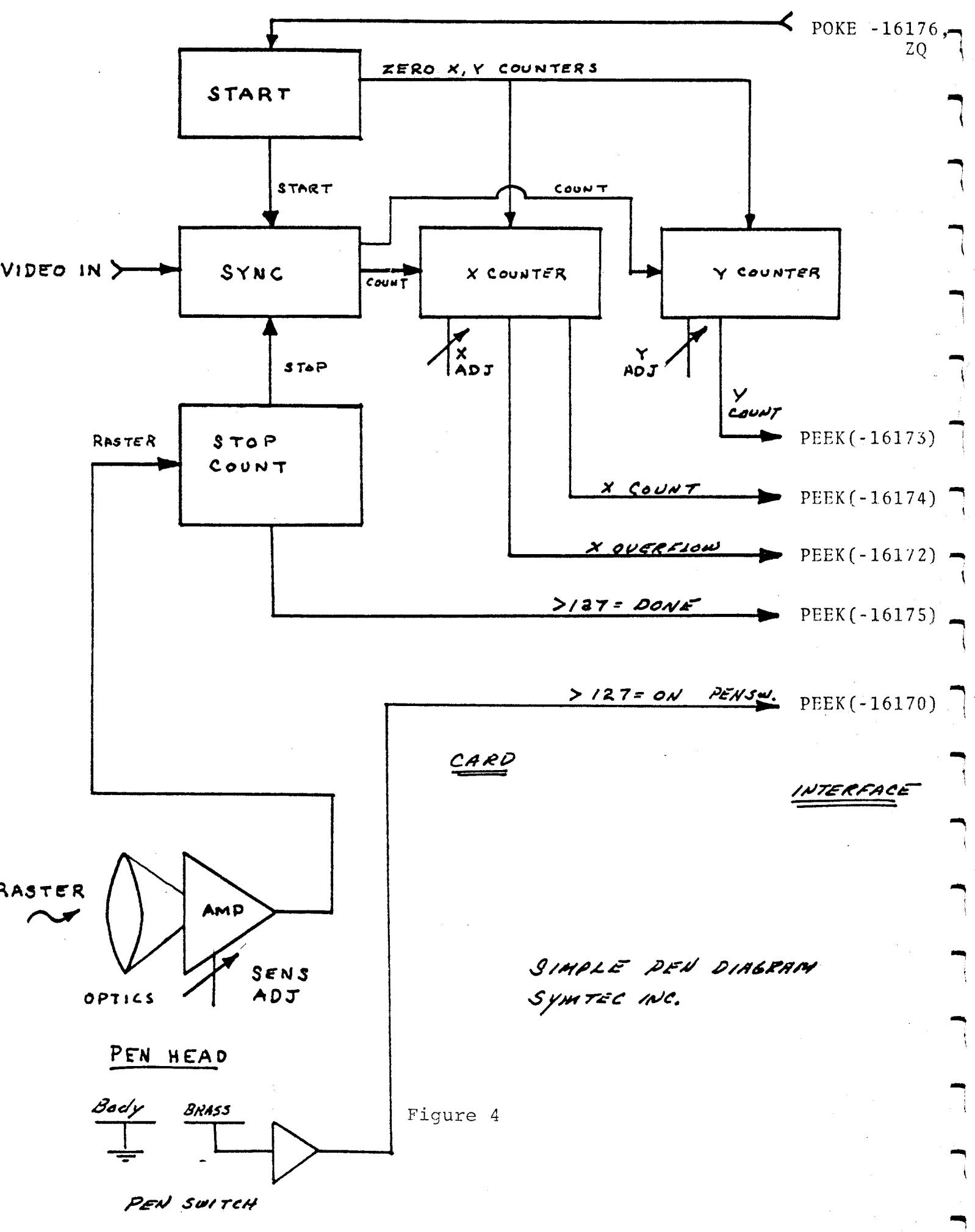
4 Y LORES = Y HIRES/4

NOTE: All division is integer division.

It can be seen that using the equations listed above and the pictorial representations of the mode in Appendix 2, one can easily go from any mode representation to any other. Using these algorithms, it sometimes may be desired to use the HIRES number subroutine and do your own conversion external to the light pen subroutine.

Theory of Pen Operation

In Figure 4 there is a simple block diagram of how the Symtec Light Pen works. The picture on your television receiver is generated by what is called a raster scan on the CRT or cathode ray tube. This means that the image is made one line at a time with the electron beam which generates this picture starting at the upper left hand corner of your screen and producing a line of picture information by proceeding to the right across your screen. The electron beam then rapidly resets without drawing on the CRT to the left hand portion of the screen and draws another line which is beneath the first. It does so again and again, in this manner, until a complete picture has been drawn. When the Light Pen is put up to the television screen, it sees an area which is roughly described in Figure 5. It sees several scan lines and several points on each scan line. This figure is many times increased in scale and this teardrop would roughly be one-quarter of an inch high and a few inches in the horizontal direction on a normal size television receiver. This teardrop pattern is due to the optics of the Light Pen. The size of the teardrop



SIMPLE PEN DIAGRAM
SYNTEC INC.

Figure 4

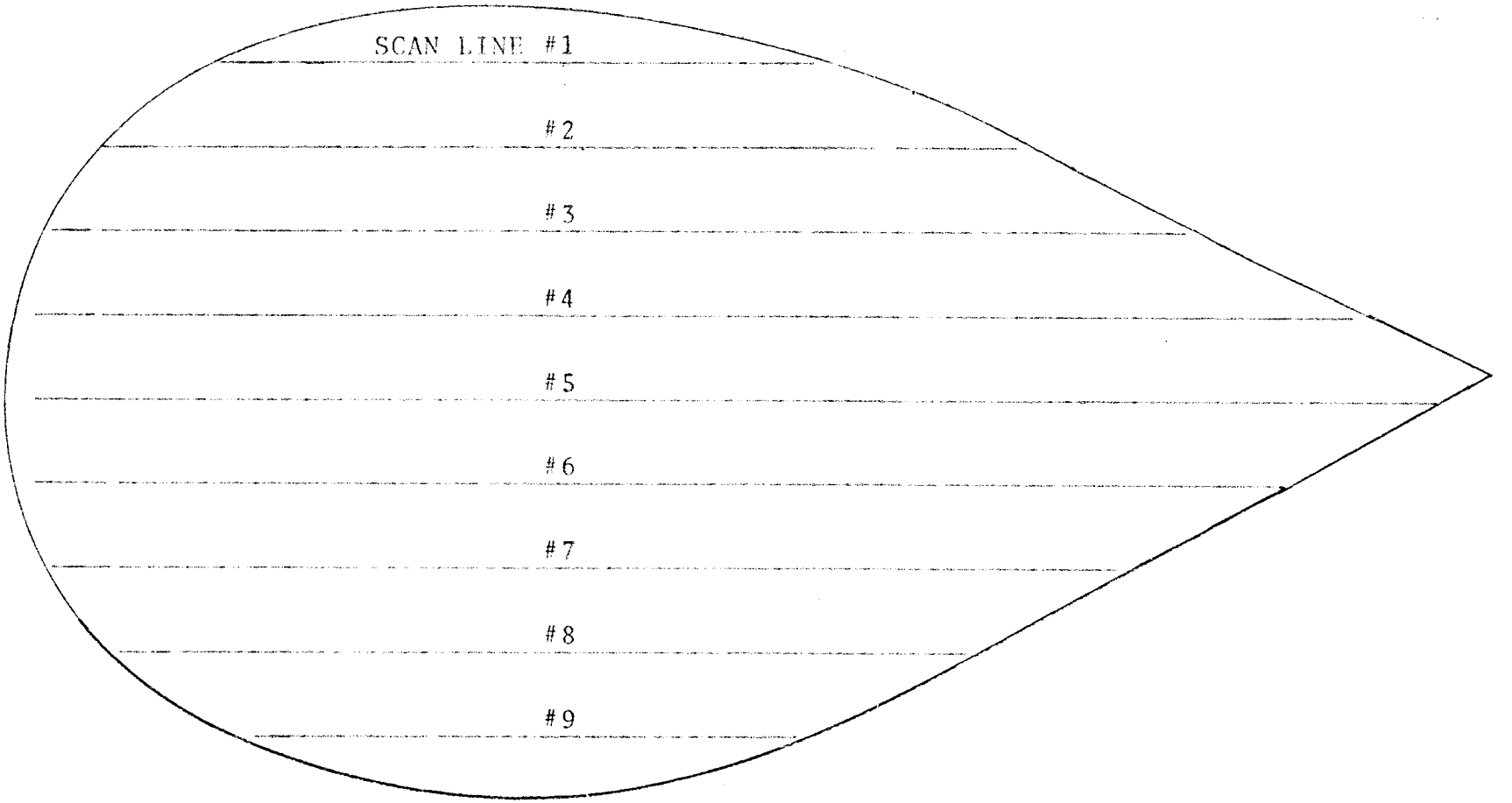


Figure 5

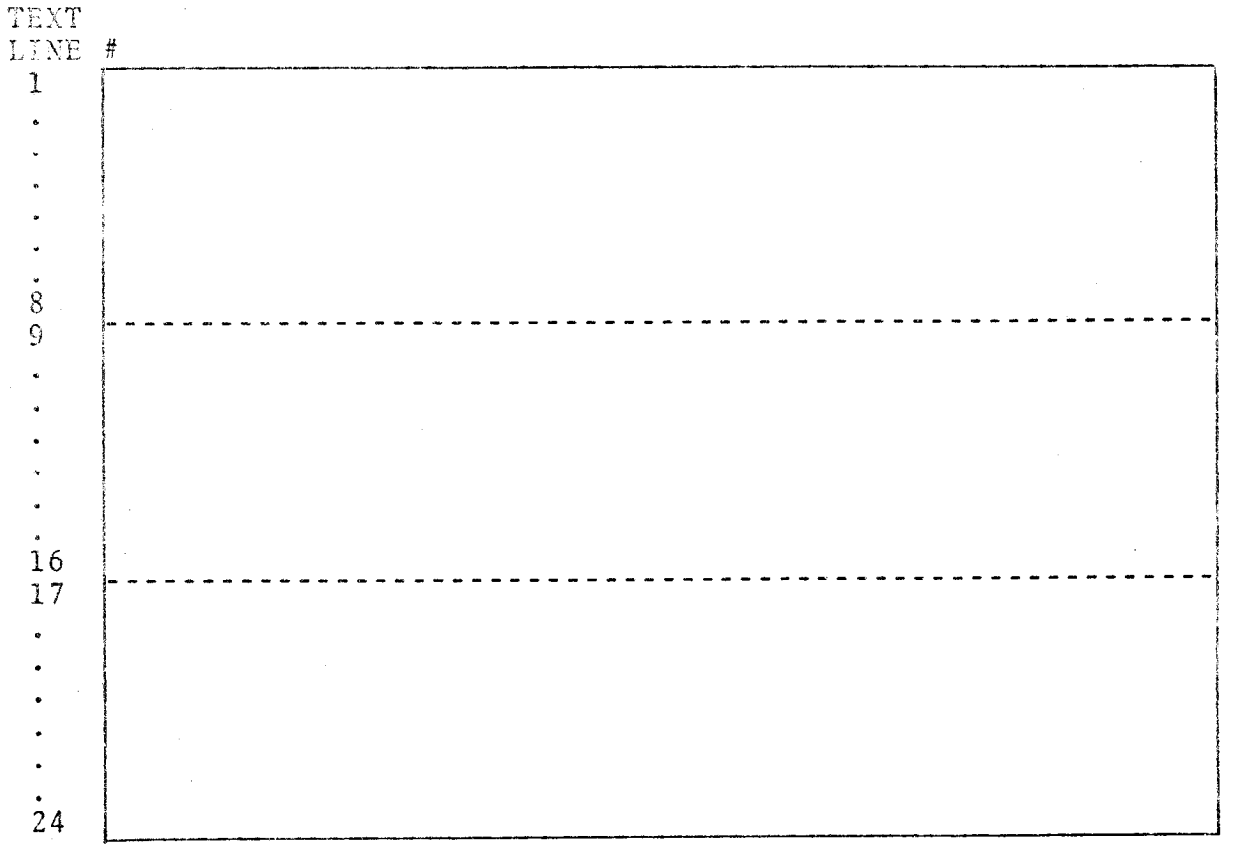
region is dependent on many things. Those are the brightness of the screen, sensitivity of the light pen setting, distance of light pen to CRT, angle of pen to CRT, background noise on television screen, the background level of the television screen, and the CRT phosphor in your television picture tube. These quantities vary from set to set. So it was necessary to devise a method to resolve a HIRES point that would be independent of the television the light pen was working with. The electronics of the light pen work as diagramed in Figure 4. POKE-ing a number, which is called ZQ in the provided subroutines into location -16176, starts the circuitry looking for a point on the television screen. It zeroes two counters, one counter counting the X position and another counting the Y position and transfers the START signal to the sync module. On every video output that goes in to your television set from the Apple II there is a pulse telling it when to begin drawing at the upper left hand corner of your screen. This is called the sync pulse on your video signal. The sync module waits until the sync pulse has been received from the video that the Apple II supplies and then starts the X and Y counters counting. They stop counting when the light pen senses a bright spot going through its field of view, which means that the electron beam is writing where the light pen is looking. You will note in the teardrop pattern of Figure 5 that if this was done, the light pen would return a location which is roughly the left hand boundary of scan line #1. This

is not the most stable position on the teardrop pattern which the light pen sees. The most stable position is the left most scan line or in other words, the position which has the minimum X coordinate. This has been taken care of in the PC card circuitry and the subroutines provided. When one value ZQ is poked into location -16176, this tells the circuitry to stop at the (ZQ + 1)th pulse. If ZQ = 0, it will stop at the first pulse, which is the left hand boundary of the first scan line. If ZQ = 1, it will stop when the light pen sees the second pulse, or another words it will stop with the coordinate of the left hand boundary of the second scan line in the teardrop pattern. In the subroutines, it can be seen that we put the ZQ variable in a for next loop, looking all the while at increasing values of the scan line. We remember the X position and Y position of the previous point that the light pen had generated. And we look to see if the new X coordinate is greater than the old X coordinate. If it is, then we know that the old X coordinate was a minimum value of X and we should return that value for X and for Y. If the new value is not greater than the old, then it says simply that we have not turned the corner as it were, in the value for X and to proceed with the value of X for the next scan or line. So simply redo the program and increment VQ. This will give us a HIRES number which is very stable. But it is still not stable enough for our purposes, so that our subroutine does this operation of providing a stable HIRES number more than once.

It loads the old stable HIRES X and HIRES Y coordinates into Z1 and Z2 and then proceeds to gather a new stable X and Y coordinate. When the new X and Y coordinate are in exact match of the old stable X and Y coordinate, then, and only then, will the subroutine return those coordinates. As a recap, you can see from the listing provided that there are actually two operations that are going on at the same time in our subroutines. 1) Finding the minimum value of ZX in order to give us a stable X and Y HIRES coordinate, and then 2) making sure that we get the same stable HIRES coordinate two times in a row before outputting that coordinate. This allows us to have a HIRES accuracy of plus or minus 1 HIRES pixel.

Text Outputting in the Non-Destructive Cursor

If you have run either the catalog select demo or one of the versions of the user modes select subroutines with text, you have seen the non-destructive cursor and have probably outputted a text string corresponding to the character which the non-destructive cursor is positioned over. To understand how these operations were accomplished, one needs to know the ways the Apple II maps screen text into its memory. Looking at Figure 6, we see the screen divided into three portions when it is in full text mode. These portions have Y coordinate in text mode of 1 through 8, 9 through 16, and 17 through 24. The Apple II provides a space in its memory for the screen text display from 1024 to 2048 in decimal. It represents each



Screen Memory Divisions

Figure 6

character by a one byte or 0 to 255 number in memory. To find out which character corresponds to which number, look in Appendix 3 under Apple II keyboard code. The first line of text with Y coordinate = 1 is written into memory starting at memory location in decimal 1024 so that in text position 11 is mapped into 1024. Position 12, that is the first row, second column, or VTAB 1, TAB 2, is written into 1025. This continues until the entire first line of text has been written into memory. The next memory location has the character representation of the character located at position 91, that is Y coordinate 9, X coordinate 1. This is the first line of the second portion of the screen. Again, that entire line is outputted starting at location 1064 and ending at location 1103 decimal. Memory then proceeds down to the first line of the third section of the screen, that is, Y coordinate 17 and X coordinate 1 through 40. After writing that line into memory you have taken up 120 memory locations. The memory then skips 8 locations so that each segment of screen memory comes in 128 byte lengths. It then proceeds to the first region of screen memory, the second line and first column, in other words, Y coordinate 2, X coordinate 1 and redoes this process of interlacing screen text into memory. This requires an 8 times 128 bytes of memory which fills the space allotted. By reading directly into memory of POKE-ing into a specific memory location, the number which corresponds to the character you wish to represent, the character will suddenly appear on

the screen. Possibly if you have seen the start of some popular programs, they immediately load their logo into this screen memory. Because they load it in in a linear manner, the logo appears by interlacing the three portions of the screen. First the top, then the middle, then the bottom, then another line at the top, another line in the middle, etc.

The alpha string ZZ$ was outputted from the subroutine by simply taking the text coordinate and converting that to the appropriate place in memory. It would then PEEK at this location and get a value. This value would be converted to the appropriate ASCII representation and then the character which corresponds to that ASCII representation would be found and assigned to ZZ$. To make the flashing non-destructive cursor, the text coordinates were found translated into screen memory coordinates. The number which was stored there was augmented by 64. This put the normal mode into a flashing mode of text. When the pen moved from that location, the normal mode was replaced by simply reading out the number in that memory location and subtracting 64, thus replacing it with its old value, so that as the non-destructive cursor passed over a letter it would not erase the letter that was on the screen. This is a simple algorithm for making a non-destructive text cursor but in appearance it is very impressive.

ADDENDUM

Applesoft and the HIRES # Machine Code program reside in the same memory space, so they cannot be used together. For this reason, another machine code program has been placed on Cassette Side #1 after the first, which is to be loaded into space 300.37FR which is vacant in Applesoft. When using this program make the following address changes:

16384	goes to	768
16500		884
16501		885
16504		888

-16170 the touch ring address is independant of this move, as you should realize.

MACHINE CODE HIRES

```
4000-   A9 FF      LDA   #$FF
4002-   8D 72 40   STA   $4072
4005-   A9 00      LDA   #$00
4007-   8D 73 40   STA   $4073
400A-   A9 FF      LDA   #$FF
400C-   8D 74 40   STA   $4074
400F-   A9 FF      LDA   #$FF
4011-   8D 75 40   STA   $4075
4014-   A9 03      LDA   #$03
4016-   8D 76 40   STA   $4076
4019-   CE 76 40   DEC   $4076
401C-   F0 53      BEQ   $4071
401E-   AD 73 40   LDA   $4073
4021-   30 3F      BMI   $4062
4023-   8D D0 C0   STA   $C0D0
4026-   AD D1 C0   LDA   $C0D1
4029-   10 FB      BPL   $4026
402B-   AD D4 C0   LDA   $C0D4
402E-   CD 74 40   CMP   $4074
4031-   B0 2F      BCS   $4062
4033-   AD D2 C0   LDA   $C0D2
4036-   CD 75 40   CMP   $4075
4039-   B0 27      BCS   $4062
403B-   8D 75 40   STA   $4075
403E-   AD D4 C0   LDA   $C0D4
4041-   8D 74 40   STA   $4074
4044-   AD D3 C0   LDA   $C0D3
4047-   8D 78 40   STA   $4078
404A-   AD 73 40   LDA   $4073
404D-   8D 77 40   STA   $4077
4050-   6D 73 40   ADC   $4073
4053-   ED 72 40   SBC   $4072
4056-   8D 73 40   STA   $4073
4059-   AD 77 40   LDA   $4077
405C-   8D 72 40   STA   $4072
405F-   4C 14 40   JMP   $4014
4062-   AD 72 40   LDA   $4072
4065-   6D 72 40   ADC   $4072
4068-   ED 73 40   SBC   $4073
406B-   8D 73 40   STA   $4073
406E-   4C 19 40   JMP   $4019
4071-   60         RTS
4072-   00         BRK
4073-   00         BRK
4074-   00         BRK
4075-   00         BRK
4076-   00         BRK
4077-   00         BRK
4078-   00         BRK
4079-   00         BRK
407A-   00         BRK
407B-   00         BRK
407C-   00         BRK
407D-   00         BRK
407E-   00         BRK
407F-   00         BRK
```

HIRES #'s

APPLESOFT II

```
]LIST
29000  REM  LIGHT PEN SUBROUTINE
29005 Z1 = 512.:Z2 = 512.
29010 Z3 = 512.:Z4 = 512.
29020  FOR ZQ = 0 TO 100
29030  POKE  - 16176,ZQ
29040 ZA =  PEEK ( - 16175)
29050  IF ZA < 128. THEN 29040
29060 ZX =  PEEK ( - 16174) + 256
     . * ( PEEK ( - 16172) > 127)

29065 ZY =  PEEK ( - 16173)
29070  IF Z3 > ZX THEN 29080
29071 ZX = Z3:ZY = Z4
29072  IF ZX = Z1 AND ZY = Z2 THEN
     RETURN
29073 Z1 = ZX:Z2 = ZY: GOTO 29020

29080 Z3 = ZX:Z4 = ZY: NEXT ZQ
29090  PRINT "ERROR IN LIGHT PEN"

29099  RETURN
29100  REM       ##################
29101  REM       #                #
29102  REM       # COPYRIGHT (C)  #
29103  REM       #      1978      #
29104  REM       #       BY       #
29105  REM       #                #
29106  REM       #    ABW CORP    #
29107  REM       #                #
29108  REM       #      AND       #
29109  REM       #                #
29110  REM       #  SYMTEC INC    #
29111  REM       #                #
29112  REM       ##################
```

INTEGER BASIC

```
>LIST
29000 REM  LIGHT PEN SUBROUTINE
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=0 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256*( PEEK
     (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN RETURN

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN
29100 REM       ##################
29101 REM       #                #
29102 REM       # COPYRIGHT (C)  #
29103 REM       #      1978      #
29104 REM       #       BY       #
29105 REM       #                #
29106 REM       #    ABW CORP    #
29107 REM       #                #
29108 REM       #      AND       #
29109 REM       #                #
29110 REM       #  SYMTEC INC    #
29111 REM       #                #
29112 REM       ##################
```

INTEGER BASIC USER MODE SELECT

```
LIST
    1 DIM ZZ$(1),ZR$(64)
    2 ZR$="@ABCDEFGHIJKLMNOPQRSTUVWXYZ
      ZZZ^Z !Z#$%&'( )*+,-./0123456789:
      ;<=>?"
29000 REM  LIGHT PEN SUBROUTINE
29003 ZV=0
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=0 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256*( PEEK
      (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN 29100

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN
29100 REM  NONDESTRUCT CURSOR AND TEXT

29103 IF ZM#4 AND (ZM#3 OR ZY<160
      ) AND (ZM#1 OR ZY<160) THEN
      29129
29106 IF ZV#0 THEN POKE ZW,Z9
29109 ZU=1024+128*((ZY/8) MOD 8)+
      40*(ZY/64)+ZX/7
29112 ZN= PEEK (ZU)
29115 IF ZT=0 THEN 29124
29118 ZS=(ZN MOD 64)+64*((ZN MOD
      64)<32)+128
29121 FOR ZI=1 TO 64: IF ASC(ZR$(
      ZI,ZI))#ZS THEN NEXT ZI
29122 ZZ$=ZR$(ZI,ZI)
29124 POKE ZU,((ZN MOD 64)+64)
29125 FOR ZI=1 TO 100: NEXT ZI
29127 ZW=ZU:ZV=1:Z9=ZN
29128 GOTO 29130
29129 IF ZV=0 THEN 29130: POKE ZW,
      Z9:ZV=0
29130 IF PEEK (-16170)<128 THEN 29005
29133 IF ZM#4 AND (ZM#3 OR ZY<160
      ) AND (ZM#1 OR ZY<160) THEN
      29136
29134 POKE ZW,Z9
29135 REM  MODE COMPENSATION
29136 ZP=0
29139 IF ZM#1 THEN 29154
29142 IF ZY<160 THEN 29154
29145 ZY=ZY/8+1
29148 ZX=ZX/7+1
29151 ZP=1
29154 IF ZM#2 AND ZM#3 THEN 29175

29157 ZX=ZX/7
29160 ZY=ZY/4
29162 CALL -1052
29163 IF ZM=2 OR ZY<40 THEN 29175

29166 ZY=ZY/2+1
29169 ZX=ZX+1
29172 ZP=1
29175 IF ZM#4 THEN 29184
29178 ZX=ZX/7+1
29181 ZY=ZY/8+1
29184 CALL -1052
29187 RETURN
29200 REM      ##################
29201 REM      #                #
29202 REM      # COPYRIGHT (C)  #
29203 REM      #      1978      #
29204 REM      #                #
29205 REM      #       BY       #
29206 REM      #                #
29207 REM      #    ABW CORP.   #
29208 REM      #                #
29209 REM      #       AND      #
29210 REM      #                #
29211 REM      #  SYMTEC INC.   #
29212 REM      #                #
29213 REM      ##################
```

```
1  DIM ZZ$(1)
29000  REM  LIGHT PEN SUBROUTINE
29005 Z1 = 512.:Z2 = 512.
29010 Z3 = 512.:Z4 = 512.
29020  FOR ZQ = 0 TO 100
29030  POKE  - 16176,ZQ
29040 ZA =  PEEK ( - 16175)
29050  IF ZA < 128. THEN 29040
29060 ZX =  PEEK ( - 16174) + 256
      * ( PEEK ( - 16172) > 127)

29065 ZY =  PEEK ( - 16173)
29070  IF Z3 > ZX THEN 29080
29071 ZX = Z3:ZY = Z4
29072  IF ZX = Z1 AND ZY = Z2 THEN
     29100
29073 Z1 = ZX:Z2 = ZY: GOTO 29020

29080 Z3 = ZX:Z4 = ZY: NEXT ZQ
29090  PRINT "ERROR IN LIGHT PEN"

29099  RETURN
29100  REM  NONDESTRUCT CURSOR
     AND TEXT
29103  IF ZM <  > 4 AND (ZM <  >
     3 OR ZY < 160) AND (ZM <  >
     1 OR ZY < 160) THEN 29129
29106  IF ZV <  > 0 THEN  POKE ZW
     ,Z9
29107 ZL =  INT (ZY / 8):ZB = 8: GOSUB
     29190
29109 ZU = 1024 + 128 *  INT (ZO)
      + 40 *  INT (ZY / 64) +  INT
     (ZX / 7)
29112 ZN =  PEEK (ZU)
29115  IF ZT = 0 THEN 29124
29116 ZL = ZN:ZB = 64: GOSUB 2919
     0
29118 Z8 = ZO + 32 * (ZN < 32)
29122 ZZ$ =  CHR$ (Z8)
29124  POKE ZU,(ZO + 64)
29125  FOR ZI = 1 TO 100: NEXT ZI

29127 ZW = ZU:ZV = 1:Z9 = ZN
29128  GOTO 29130
29129  IF ZV = 0 THEN 29130: POKE
     ZW,Z9:ZV = 0
29130  IF  PEEK ( - 16170) < 128 THEN
     29005
29133  IF ZM <  > 4 AND (ZM <  >
     3 OR ZY < 160) AND (ZM <  >
     1 OR ZY < 160) THEN 29136
29134  POKE ZW,Z9
29135  REM  MODE COMPENSATION
29136 ZP = 0
29139  IF ZM <  > 1 THEN 29154
29142  IF ZY < 160 THEN 29154
29145 ZY =  INT (ZY / 8) + 1
29148 ZX =  INT (ZX / 7) + 1
29151 ZP = 1
29154  IF ZM <  > 2 AND ZM <  > 3
     THEN 29175
29157 ZX =  INT (ZX / 7)
29160 ZY =  INT (ZY / 4)
29162  CALL  - 1052
29163  IF ZM = 2 OR ZY < 40 THEN
     29175
29166 ZY =  INT (Y / 2) + 1
29169 ZX = ZX + 1
29172 ZP = 1
29175  IF ZM <  > 4 THEN 29184
29178 ZX =  INT (ZX / 7)
29181 ZY =  INT (ZY / 8) + 1
29184  CALL  - 1052
29187  RETURN
29190  REM   ZO=ZL MOD ZB
29191 ZO = ZL - (ZB *  INT (ZL /
     ZB))
29192  RETURN
29200  REM      ##################
29201  REM      #                #
29202  REM      # COPYRIGHT (C)  #
29203  REM      #      1978      #
29205  REM      #                #
29206  REM      #    ABW CORP    #
29207  REM      #                #
29208  REM      #      AND       #
29209  REM      #                #
29210  REM      #  SYNTEC INC.   #
29211  REM      #                #
29212  REM      ##################
```

```
>LIST
    1 DIM Z(375),IK(375),EC(3),DD$
      (100),CC$(160)
    2 AP=0
    9 TEXT : CALL -936
   10 VTAB 4: TAB 9: PRINT "1 OR 2 MAN
       CHECKERS FOR": TAB 13: PRINT
      "SYMTEC LIGHT PEN"
   11 VTAB 10: TAB 10: PRINT "SOFTWARE
       COPYRIGHTED": VTAB 12: TAB
      19: PRINT "BY": VTAB 14: TAB
      9
   12 PRINT "ABW CORP. & SYMTEC INC."
      : VTAB 17: TAB 4: PRINT "   POIN
      T PEN AT X TO CONTINUE": POKE
      1123,24
   13 GOSUB 29000: CALL -1052: FOR
      GH=1 TO 100: NEXT GH
   14 CALL -936: PRINT "THE RULES ARE
      LIKE REGULAR CHECKERS     EXCEPT
      :-YOU DON'T HAVE TO JUMP-AND"

   15 PRINT "LANDING ON YOUR KING ROW
      ENDS YOUR MOVE.PLAYER #1 STARTS
      AT THE TOP, PLAYER #2  AT THE BO
      TTOM."
   16 PRINT "TO MOVE A MAN, POINT TO I
      T AND TOUCH THERING UNTIL THE AP
      PLE BEEPS;THEN RELEASE.THAT PIEC
      E WILL FLASH";
   17 PRINT "(IF YOU CHANGE YOURMIND,T
      OUCH THE RING AGAIN BEFORE YOU
        MOVE THIS PIECE).  NOW POINT T
      O";
   18 PRINT " WHERE   YOU WANT TO MOVE
       AND TOUCH THE RING.    WHEN YOU
       HEAR THE BEEP YOUR TURN IS OVER
      "
   19 VTAB 17: TAB 4: PRINT "   POINT
      PEN AT X TO CONTINUE": POKE
      1123,24
   20 GOSUB 29000: CALL -1052: FOR
      GH=1 TO 100: NEXT GH
   25 CALL -936: GR : COLOR=15: FOR
      GY=0 TO 39: HLIN 0,39 AT GY:
       NEXT GY: COLOR=6: VLIN 4,35
       AT 9: HLIN 24,35 AT 4: VLIN
      4,19 AT 35
   26 HLIN 24,35 AT 19: VLIN 19,35
       AT 24: HLIN 24,35 AT 35: PRINT
      "SELECT 1 OR 2 MAN GAME AND ENTE
      R USING  TOUCH RING"
   27 IF PEEK (-16170)>127 THEN 28
      : GOTO 27
   28 GOSUB 29000:PN=ZX/140+1: CALL
      -1052
   30 FOR LY=1 TO 16
   31 FOR LX=1 TO 16
   32 Z(LX+LY*20)=0: IF (LX+LY) MOD
      2=0 THEN 36
   33 IF LY<8 THEN Z(LY*20+LX)=1
   34 IF LY>9 THEN Z(LY*20+LX)=2
   35 IF LY=8 OR LY=9 THEN Z(LY*20
      +LX)=0
   36 IK(LY*20+LX)=-20
   37 IF LX<5 OR LX>12 OR LY<5 OR
      LY>12 THEN Z(LY*20+LX)=3
   38 NEXT LX
   39 NEXT LY
   40 CC$="BLACK      MAGENTA   DARK BL
      UE PURPLE    DARK GREENGREY
       MED. BLUE LIGHT BLUE"
   41 GOSUB 2900: GR :CC$(81)="BROWN
          ORANGE     GREY       PINK
        GREEN      YELLOW     AQUA
      WHITE      "
   43 CZ=0:DD$="DARK SQUARE COLOR ISLI
      GHT SQ. COLOR IS  PLAYER #1 COLO
      R IS  PLAYER #2 COLOR IS
       "
   44 FOR I=0 TO 15
   45 FOR Y=0 TO 39
   46 COLOR=I
   47 X=I*2+4
   48 HLIN X,X+1 AT Y
   50 NEXT Y
   52 NEXT I
   54 COLOR=6
   56 HLIN 3,36 AT 0
   58 HLIN 3,36 AT 39
   60 VLIN 0,39 AT 3
   62 VLIN 0,39 AT 36
   63 FOR EE=0 TO 3
   64 GOSUB 29000:CZ=C
   65 IF ZX<28 THEN 64
   66 C=((ZX/7)-4)/2
   67 VTAB 22
   68 IF CZ<>C THEN PRINT DD$(EE*20
      +1,EE*20+20),CC$(C*10+1,C*10
      +10)
   69 IF PEEK (-16170)>128 THEN 72

   70 GOTO 64
   72 CALL -1052
   73 FOR IZ=1 TO 100: NEXT IZ
   74 EC(EE)=C
   75 NEXT EE
   77 GR : CALL -936: GOSUB 1300
   78 FOR LX=5 TO 12
   79 FOR LY=5 TO 12
   80 IF (LX+LY) MOD 2=0 THEN 83
   81 QQ=LY*20+LX
   82 GOSUB 1100: GOSUB 1000
   83 NEXT LY
   84 NEXT LX
```

```
100 FOR PT=1 TO PN
102 CALL -936
103 PRINT "PLAYER #";PT;" TURN"

105 FOR IZ=1 TO 100: NEXT IZ
106 IF PEEK (-16170)<128 THEN 106

109 CALL -1052
110 QP=0
111 QB=0
112 GOSUB 1400:QM=DX+DY*20+84:QS=
    QM
114 IF Z(QM)#PT THEN 112
116 QQ=QM: GOSUB 1200
118 GOSUB 1400:QQ=DX+DY*20+84: IF
    QQ=QS AND QP=1 AND PEEK (-16170
    )>128 THEN
    QB=1
119 IF QQ=QM AND QP=1 AND PEEK
    (-16170)>127 THEN 152:QP=1
120 IF Z(QQ)#0 THEN 118
122 QA=(QQ-QM): IF ABS (QA)#19 AND
    ABS (QA)#21 AND ABS (QA)#38
     AND ABS (QA)#42 THEN 118
124 QV=(QQ+QM)/2: IF SGN (QA)#(
    3-PT*2) AND IK(QM)#20 THEN
    118
126 IF ABS (QA)>21 AND Z(QV)#(1
    +(PT MOD 2)) THEN 118
128 QH=QQ:QQ=QM: GOSUB 1100:Z(QH)
    =Z(QQ):IK(QH)=IK(QQ):Z(QQ)=
    0:IK(QQ)=-20:QQ=QH
130 IF ABS (QA)<22 THEN 134
132 QQ=QV: GOSUB 1100:Z(QQ)=0:IK(
    QQ)=-20:QQ=QH
134 GOSUB 1200:QM=QQ
136 IF PEEK (-16170)<128 THEN 118

152 CALL -1052: IF QB#1 THEN 153
    : VTAB 21: PRINT "MOVE AGAIN
         ": FOR QW=1 TO 250: NEXT
    QW: GOTO 102
153 GOSUB 1700
154 GOSUB 1500
155 IF PN=1 THEN GOSUB 900
156 PRINT "HI THER"
157 NEXT PT
160 GOTO 100

900 REM  COMPUTER MOVE
902 VTAB 21: PRINT "I'M THINKING
    "
903 JUMP=0
904 FOR PP=1 TO 4
906 FOR LY=5 TO 12
908 FOR LX=5 TO 12
909 IF (LX+LY) MOD 2=0 THEN 922

910 Q1=(LX+LY*20)
911 IF Z(Q1)#2 THEN 922
912 IF PP=1 THEN GOSUB 1800
914 IF PP=2 THEN GOSUB 2000
916 IF PP=3 THEN GOSUB 2400
918 IF PP=4 THEN GOSUB 2600
920 IF JUMP=1 THEN 928
922 NEXT LX
924 NEXT LY
926 NEXT PP
928 IF JUMP#0 THEN 940
930 VTAB 21: PRINT "I CAN'T MOVE"

932 RETURN
940 IF PP#1 THEN 950
942 QQ=Q1: GOSUB 1200: GOSUB 1100
    :Z(W4)=2:IK(W4)=IK(Q1):IK(Q1)
    =-20:Z(Q1)=0
943 QQ=W1: GOSUB 1100:IK(W1)=-20
    :Z(W1)=0
944 QQ=W2:Z(QQ)=Z(W4):IK(QQ)=IK(
    W4): GOSUB 1200: GOSUB 1100
    :Z(QQ)=0:IK(QQ)=-20
945 QQ=W3: GOSUB 1100:IK(W3)=-20
    :Z(W3)=0
946 QQ=W4: GOSUB 1000
948 GOTO 970
950 IF PP#2 THEN 960
952 QQ=Q1: GOSUB 1200: GOSUB 1100
    :IK(W2)=IK(Q1):Z(W2)=2:Z(Q1)
    =0:IK(Q1)=-20
954 QQ=W1: GOSUB 1100:IK(W1)=-20
    :Z(W1)=0
956 QQ=W2: GOSUB 1000
958 GOTO 970
960 QQ=Q1: GOSUB 1200: GOSUB 1100
    :Z(W1)=2:IK(W1)=IK(Q1):Z(Q1)
    =0:IK(Q1)=-20
962 QQ=W1: GOSUB 1000
970 GOSUB 1700
972 GOSUB 1500
974 RETURN
```

```
1000 REM  DRAW
1003 XX=(QQ MOD 20-5)*5
1006 YY=(QQ/20-5)*5+2
1009 IF Z(QQ)=1 THEN COLOR=EC(2)

1012 IF Z(QQ)=2 THEN COLOR=EC(3)

1015 FOR ID=-2 TO 2
1018 HLIN (XX+ ABS (ID)),(XX+4- ABS
     (ID)) AT (YY+ID)
1021 NEXT ID
1024 IF IK(QQ)<0 THEN 1036
1027 IF Z(QQ)=1 THEN COLOR=EC(3)

1030 IF Z(QQ)=2 THEN COLOR=EC(2)

1033 PLOT (XX+2),YY
1036 RETURN
1100 REM  ERASE
1102 COLOR=EC(0)
1103 XX=(QQ MOD 20-5)*5
1106 YY=(QQ/20-5)*5
1109 FOR IE=0 TO 4
1112 HLIN XX,XX+4 AT YY+IE
1115 NEXT IE
1118 RETURN
1200 REM  FLASH
1203 FOR II=1 TO 5
1206 GOSUB 1100
1207 FOR IZ=1 TO 1: NEXT IZ
1209 GOSUB 1000
1210 FOR IZ=1 TO 1: NEXT IZ
1212 NEXT II
1215 RETURN
1300 REM  WHITE SQUARES
1303 EZ=EC(0)
1305 EC(0)=EC(1)
1306 FOR LX=5 TO 12
1309 FOR LY=5 TO 12
1312 IF (LX+LY) MOD 2#0 THEN 1318

1315 QQ=LX+LY*20: GOSUB 1100
1318 NEXT LY
1321 NEXT LX
1324 EC(0)=EZ
1327 RETURN
1400 REM  DETECT
1403 GOSUB 29000
1406 DX=ZX/7:DY=ZY/4
1408 DX=DX/5:DY=DY/5
1409 DX=DX+1:DY=DY+1
1412 IF (DY+DX) MOD 2=0 THEN 1403

1415 RETURN
1500 REM  WIN LOSE
1502 AA=0:BB=0
1503 FOR TX=5 TO 12
1506 FOR TY=5 TO 12
1509 QQ=TY*20+TX
1512 IF Z(QQ)=1 THEN AA=AA+1
1515 IF Z(QQ)=2 THEN BB=BB+1
1518 NEXT TY
1521 NEXT TX
1524 IF AA#0 THEN 1530
1527 TEXT : CALL -936: PRINT "PLAYER
          #2 WINS": GOSUB 1600

1530 IF BB#0 THEN 1536
1533 TEXT : CALL -936: PRINT "PLAYER
          #1 WINS": GOSUB 1600

1536 RETURN
1600 REM  WANT TO PLAY AGAIN
1603 FOR IZ=1 TO 5
1606 CALL -1052
1609 NEXT IZ
1612 PRINT "WANT TO PLAY AGAIN?"

1613 VTAB 15
1615 PRINT "     YES                NO"

1617 IF PEEK (-16170)<128 THEN 1617

1618 GOSUB 29000
1621 IF ZX>100 THEN 30000
1622 POP
1623 POP
1624 CALL -1052
1625 GOTO 2
1627 RETURN
1700 REM  KING?
1702 IF IK(QQ)=20 THEN 1709
1703 IF (QQ/20=12 AND Z(QQ)=1) OR
     (QQ/20=5 AND Z(QQ)=2) THEN
     IK(QQ)=20
1706 GOSUB 1000
1709 RETURN
1800 REM  MOVE 2 JUMP
1802 JUMP=0
1804 Z1=-21:Z2=-42:Z3=-63:Z4=-84

1806 GOSUB 1900
1808 IF JUMP=1 THEN RETURN
1810 Z1=-21:Z2=-42:Z3=-61:Z4=-80

1812 GOSUB 1900
1814 IF JUMP=1 THEN RETURN
1816 Z1=-19:Z2=-38:Z3=-57:Z4=-76
```

```
1818 GOSUB 1900
1820 IF JUMP=1 THEN RETURN
1822 Z1=-19:Z2=-38:Z3=-59:Z4=-80

1824 GOSUB 1900
1826 IF JUMP=1 THEN RETURN
1828 IF IK(Q1)#20 THEN RETURN
1830 Z1=-21:Z2=-42:Z3=-23:Z4=-4
1832 GOSUB 1900
1834 IF JUMP=1 THEN RETURN
1836 Z1=-19:Z2=-38:Z3=-17:Z4=4
1838 GOSUB 1900
1840 IF JUMP=1 THEN RETURN
1842 Z1=21:Z2=42:Z3=63:Z4=84
1844 GOSUB 1900
1846 IF JUMP=1 THEN RETURN
1848 IF IK(Q1)#20 THEN RETURN
1850 Z1=21:Z2=42:Z3=61:Z4=80
1852 GOSUB 1900
1854 IF JUMP=1 THEN RETURN
1856 Z1=21:Z2=42:Z3=23:Z4=4
1858 GOSUB 1900
1860 IF JUMP=1 THEN RETURN
1862 Z1=19:Z2=38:Z3=57:Z4=76
1864 GOSUB 1900
1866 IF JUMP=1 THEN RETURN
1868 Z1=19:Z2=38:Z3=59:Z4=80
1870 GOSUB 1900
1872 IF JUMP=1 THEN RETURN
1874 Z1=19:Z2=38:Z3=17:Z4=-4
1876 GOSUB 1900
1880 RETURN
1900 REM  2 JUMP DETECT
1902 W1=Q1+Z1:W2=Q1+Z2:W3=Q1+Z3:
     W4=Q1+Z4
1904 IF Z(W1)=1 AND Z(W2)=0 AND
     Z(W3)=1 AND Z(W4)=0 THEN JUMP=
     1
1906 RETURN
2000 REM  MOVE 1 JUMP
2002 JUMP=0
2004 Z1=-21:Z2=-42
2006 GOSUB 2100
2008 IF JUMP=1 THEN RETURN
2010 Z1=-19:Z2=-38
2012 GOSUB 2100
2014 IF JUMP=1 THEN RETURN
2016 IF IK(Q1)#20 THEN RETURN
2018 Z1=21:Z2=42
2020 GOSUB 2100
2022 IF JUMP=1 THEN RETURN
2024 Z1=19:Z2=38
2026 GOSUB 2100
2028 RETURN
2100 REM  I JUMP DETECT
2102 W1=Q1+Z1:W2=Q1+Z2
2104 IF Z(W1)=1 AND Z(W2)=0 THEN
     JUMP=1
2106 RETURN
2400 REM  CLEAR MOVE
2402 JUMP=0
2403 IF IK(Q1)#20 THEN 2418
2404 Z1=21:Z2=42:KK=1
2406 GOSUB 2500
2408 IF JUMP=1 THEN RETURN
2410 Z1=19:Z2=38
2412 GOSUB 2500
2414 IF JUMP=1 THEN RETURN
2418 Z1=-21:Z2=-42:KK=0
2420 GOSUB 2500
2422 IF JUMP=1 THEN RETURN
2424 Z1=-19:Z2=-38
2426 GOSUB 2500
2428 RETURN
2500 REM  CLEAR DETECT
2502 W1=Q1+Z1:W2=Q1+Z2
2504 IF Z(W1)=0 AND Z(W2)#1 THEN
     JUMP=1
2506 RETURN
2600 REM   ANY MOVE
2602 JUMP=0
2604 Z1=-21
2606 GOSUB 2700
2608 IF JUMP=1 THEN RETURN
2610 Z1=-19
2612 GOSUB 2700
2614 IF JUMP=1 THEN RETURN
2616 IF IK(Q1)#20 THEN RETURN
2618 Z1=21
2620 GOSUB 2700
2622 IF JUMP=1 THEN RETURN
2624 Z1=19
2626 GOSUB 2700
2628 RETURN
2700 REM   ANY DETECT
2702 W1=Q1+Z1
2704 IF Z(W1)=0 THEN JUMP=1
2706 RETURN
2900 REM  PRINT
2901 CALL -936: PRINT "POINT PEN AT C
     OLOR MENU AND TOUCH RING": RETURN
```

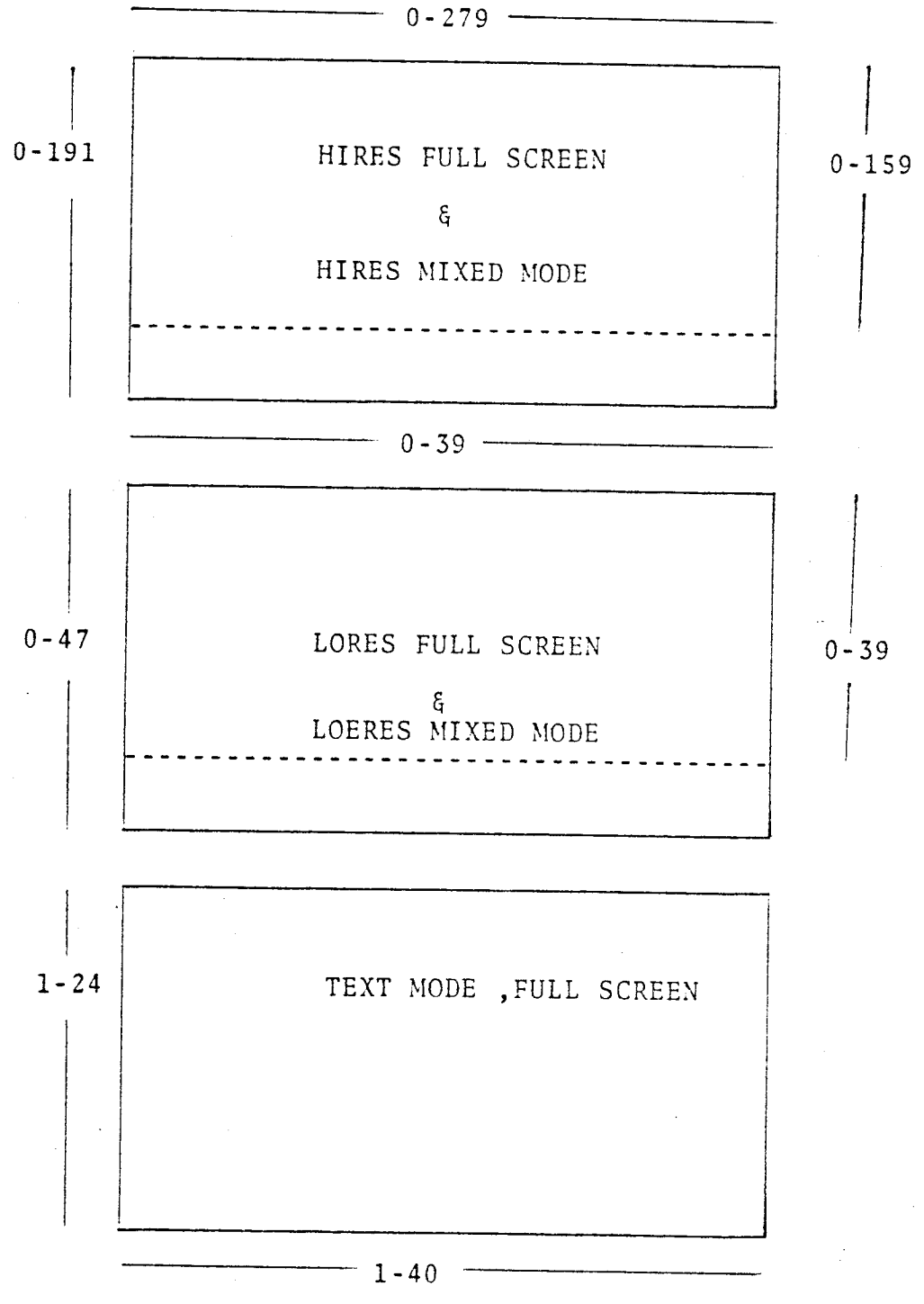
```
29000 REM  LIGHT PEN SUBROUTINE
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=1 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256*( PEEK
      (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN RETURN

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN
29100 REM     ################
29101 REM     #              #
29102 REM     # COPYRIGHT (C)#
29103 REM     #     1978     #
29104 REM     #              #
29105 REM     #   ABW CORP   #
29106 REM     #              #
29107 REM     #     AND      #
29108 REM     #              #
29109 REM     #  SYMTEC INC  #
29110 REM     #              #
29111 REM     ################
30000 END
```

"C" PROGRAM
CATALOG SELECT

```
>LIST
    0 TEXT : CALL -936: PRINT : PRINT
      "POINT LIGHT PEN AT FILE NAME FI
      RST      LETTER AND TOUCH RING"
      : PRINT "CATALOG"
    1 DIM ZZ$(40),ZR$(64)
    2 ZR$="@ABCDEFGHIJKLMNOPQRSTUVWXYZ
      ZZZ↑Z !Z$$%&'( )%+,-./0123456789:
      ;<=>?"
    3 ZT=1:ZM=4
   14 GOSUB 29000
   15 ZK=((ZY-1) MOD 8)%128+(40%(
      (ZY-1)/8))+1024+ZX
   17 ZT=((ZY-1) MOD 8)%128+(40%(
      (ZY-1)/8))+1063
   18 BL=0
   19 FOR Q0=ZK TO ZT
   21 FOR ZI=1 TO 64
   23 IF ASC(ZR$(ZI,ZI))$ PEEK (Q0)
       THEN NEXT ZI
   24 IF BL=1 AND ZI=33 THEN 29:BL=
      0: IF ZI=33 THEN BL=1
   25 ZZ$( LEN(ZZ$)+1)=ZR$(ZI,ZI)

   27 NEXT Q0
   28 IF ZZ$$"APPLESOFT" THEN 30:
       PRINT "FP": GOTO 31
   29 IF ZZ$$"APPLESOFT " THEN 30
      : PRINT "FP": GOTO 31
   30 PRINT "RUN ";ZZ$
   31 END
29000 REM  LIGHT PEN SUBROUTINE
29003 ZV=0
29005 Z1=512:Z2=512
29010 Z3=512:Z4=512
29020 FOR ZQ=0 TO 100
29030 POKE -16176,ZQ
29040 ZA= PEEK (-16175)
29050 IF ZA<128 THEN 29040
29060 ZX= PEEK (-16174)+256%( PEEK
      (-16172)>127)
29065 ZY= PEEK (-16173)
29070 IF Z3>ZX THEN 29080
29071 ZX=Z3:ZY=Z4
29072 IF ZX=Z1 AND ZY=Z2 THEN 29100

29073 Z1=ZX:Z2=ZY: GOTO 29020
29080 Z3=ZX:Z4=ZY: NEXT ZQ
29090 PRINT "ERROR IN LIGHT PEN"
29099 RETURN
29100 REM  NONDESTRUCT CURSOR AND TEXT

29103 IF ZM$4 AND (ZM$3 OR ZY<160
      ) AND (ZM$1 OR ZY<160) THEN
      29129
29106 IF ZV$0 THEN POKE ZW,Z9
29109 ZU=1024+128%((ZY/8) MOD 8)+
      40%(ZY/64)+ZX/7
29112 ZN= PEEK (ZU)
29115 IF ZT=0 THEN 29124
29118 ZS=(ZN MOD 64)+64%((ZN MOD
      64)<32)+128
29121 FOR ZI=1 TO 64: IF ASC(ZR$(
      ZI,ZI))$ZS THEN NEXT ZI
29122 ZZ$=ZR$(ZI,ZI)
29124 POKE ZU,((ZN MOD 64)+64)
29125 FOR ZI=1 TO 100: NEXT ZI
29127 ZW=ZU:ZV=1:Z9=ZN
29128 GOTO 29130
29129 IF ZV=0 THEN 29130: POKE ZW,
      Z9:ZV=0
29130 IF PEEK (-16170)<128 THEN 29005

29133 IF ZM$4 AND (ZM$3 OR ZY<160
      ) AND (ZM$1 OR ZY<160) THEN
      29136
29134 POKE ZW,Z9
29135 REM  MODE COMPENSATION
29136 ZP=0
29139 IF ZM$1 THEN 29154
29142 IF ZY<160 THEN 29154
29145 ZY=ZY/8+1
29148 ZX=ZX/7+1
29151 ZP=1
29154 IF ZM$2 AND ZM$3 THEN 29175

29157 ZX=ZX/7
29160 ZY=ZY/4
29162 CALL -1052
29163 IF ZM=2 OR ZY<40 THEN 29175

29166 ZY=ZY/2+1
29169 ZX=ZX+1
29172 ZP=1
29175 IF ZM$4 THEN 29184
29178 ZX=ZX/7+1
29181 ZY=ZY/8+1
29184 CALL -1052
29187 RETURN
29200 REM      $$$$$$$$$$$$$$$$$
29201 REM      $                 $
29202 REM      $ COPYRIGHT (C) $
29203 REM      $      1978       $
29204 REM      $                 $
29205 REM      $                 $
29206 REM      $    ABW CORP     $
29207 REM      $                 $
29208 REM      $       AND       $
29209 REM      $                 $
29210 REM      $  SYNTEC INC     $
29211 REM                        $
29212 REM      $$$$$$$$$$$$$$$$$
```

APPENDIX 2

LORES, HIRES, and Text Maps


0-279
0-191
HIRES FULL SCREEN
&
HIRES MIXED MODE
0-159
0-39
0-47
LORES FULL SCREEN
&
LOERES MIXED MODE
0-39
1-24
TEXT MODE ,FULL SCREEN
1-40

APPLE II Keyboard Character Codes

Decimal	Character	Decimal	Character
0	@	32	SPACE
1	A	33	!
2	B	34	"
3	C	35	#
4	D	36	$
5	E	37	%
6	F	38	&
7	G	39	'
8	H	40	(
9	I	41	)
10	J	42	*
11	K	43	+
12	L	44	,
13	M	45	-
14	N	46	.
15	O	47	/
16	P	48	0
17	Q	49	1
18	R	50	2
19	S	51	3
20	T	52	4
21	U	53	5
22	V	54	6
23	W	55	7
24	X	56	8
25	Y	57	9
26	Z	58	:
27	[	59	;
28	\	60	<
29	]	61	=
30	^	62	>
31	_	63	?

Character codes range from 0-255. Inverted, black-on-white characters use the given decimal numbers. For flashing characters add 64. For normal characters add 128 or 192.

APPLE II Keyboard Character Codes

Decimal	Character	Decimal	Character
0	@	32	SPACE
1	A	33	!
2	B	34	"
3	C	35	#
4	D	36	$
5	E	37	%
6	F	38	&
7	G	39	'
8	H	40	(
9	I	41	)
10	J	42	*
11	K	43	+
12	L	44	,
13	M	45	-
14	N	46	.
15	O	47	/
16	P	48	0
17	Q	49	1
18	R	50	2
19	S	51	3
20	T	52	4
21	U	53	5
22	V	54	6
23	W	55	7
24	X	56	8
25	Y	57	9
26	Z	58	:
27	[	59	;
28	\	60	<
29	]	61	=
30	^	62	>
31	_	63	?

Character codes range from 0-255. Inverted, black-on-white characters use the given decimal numbers. For flashing characters add 64. For normal characters add 128 or 192.

ASCII Character Codes
INTEGER BASIC

Decimal	Character	Decimal	Character
192	@	224	SPACE
193	A	225	!
194	B	226	"
195	C	227	#
196	D	228	$
197	E	229	%
198	F	230	&
199	G	231	
200	H	232	(
201	I	233	)
202	J	234	*
203	K	235	+
204	L	236	,
205	M	237	-
206	N	238	.
207	O	239	/
208	P	240	0
209	Q	241	1
210	R	242	2
211	S	243	3
212	T	244	4
213	U	245	5
214	V	246	6
215	W	247	7
216	X	248	8
217	Y	249	9
218	Z	250	:
219	[	251	;
220		252	
221	]	253	=
222		254	
223	_	255	?

ASCII Character Codes
APPLESOFT II

Decimal	Character	Decimal	Character
32	SPACE	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(	72	H
41	)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;	91	[
60	<	92	\
61	=	93	]
62	>	94	^
63	?	95	_

APPENDIX 4

Changing the Light Pen to a Different Slot

Moving Symtec Light Pen PC card to slot other than slot #5. The Symtec Light Pen has been designed and all programs written for its placement in slot #5. Though it is possible to place it in any unoccupied slot, except slot #0 in the Apple II main frame. The addresses for the PEEKS and POKES in the subroutines must be changed accordingly. These can be changed simply by the algorithm below.

new address = old address + (new slot #5 - 5) X 16

As an example of this algorithm, let us assume that we move the light pen to slot #3. Three minus 5 times 16 is minus 32 so that the address for POKing ZQ which is -16172 would be changed to -16208. If you are using the machine code version of the subroutine, it is also necessary to change the addresses it has in the same manner.

All software enquiries can be addressed to:

ABW Corp.
1929 Plymouth Road
Suite 5012
Ann Arbor, Michigan 48105
(313) 665-5203

All other enquiries should be addressed to:

Symtec Inc.
P.O. Box 462
Farmington, Michigan 48024
(313) 358-2999

NOTE: Complete and return SYMTEC WARRANTY registration card to be put on our mailing list for products or system updates.

LIMITED WARRANTY

Implied warranties limited to duration of express warranty (some states do not allow limitations on how long an implied warranty lasts, so the above may not apply to you).

To the original purchaser only, Symtec Inc. warrants the Symtec Light Pen, and all components therein contained, to be free from defects in materials and/or workmanship for a period of one (1) year from the date of purchase. In the event of malfunction, or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product to Symtec, Inc. at P.O. Box 462, Farmington, Michigan 48024 (postage paid) "Attention Warranty Claims Department," Symtec Inc. will, at its option, repair or replace said products or components, to whatever extent deemed necessary, to restore said product to proper operating condition. All such repairs or replacements shall be rendered by Symtec Inc. without charge for parts or labor when the product is return within ninety (90) days of the date of purchase. During the period after ninety (90) days but before one (1) year from date of purchase, all repairs or replacements will be rendered without charge for parts. Labor will be charged at the rate in effect as of the time of such repair.

The responsibility for the failure of the Symtec Inc Light Pen product, or component thereof, which at the discretion of Symtec Inc., shall have resulted from accident, abuse,

or misapplication of the product, shall be assumed by the customer, and Symtec Inc. shall assume no liability as a consequence of such events under the terms of this warranty. Further, the limited warranty provided with this Symtec Inc. product also shall be void if customer causes or permits the Symtec product to be repaired or modified by anyone other than the Symtec Inc. warranty service center.

While every effort on the part of Symtec Inc. has been made to provide clear and accurate technical information on the application of its products, Symtec Inc. assumes no liability in any events which may arise from the use of said technical information.

This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

This warranty is in lieu of all other express warranties which now or hereafter might otherwise arise with respect to this product. Any and all implied warranties, including the warranties of mercantability and fitness for particular purpose, shall have no greater duration than the duration period for the express written warranty applicable to this product, as shown above, and shall terminate automatically at the expiration of such duration period. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. No action shall be brought for breach of any implied or express warranty after one year subsequent to the expiration of the duration

period of the express written warranty.

Incidental and consequential damages caused by malfunction, defect or otherwise, and with respect to breach of any express or implied warranty, are not the responsibility of Symtec Inc. and, to the extent permitted by law, are hereby excluded both for property damage and, to the extent not unconscionable, for personal injury damage. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Symtec Inc.
P.O. Box 462
Farmington, Michigan 48024
(313) 358-2999

SYMTEC LIGHT PEN WARRANTY REGISTRATION CARD

Serial Number _____ Name _____

Model Number _____ Address _____

Purchase Date _____ City _____

Purchased From _____ State _____ Zip _____
(Name of Store)

_____ Telephone Number _____
(Address)

_____

Symtec Inc.
P.O. Box 462
Farmington, Michigan 48024
(313) 358-2999